

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

на тему

**Тренажер з теми «Математичні основи теорії алгоритмів» дистанційного
навчального курсу «Теорія програмування» та розробка його програмного
забезпечення**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Будяй Віталій Олексійович

_____ «___» _____ 2021р.
(підпис)

Науковий керівник к.ф.-м.н., доц. Черненко Оксана Олексіївна

_____ «___» _____ 2021р.
(підпис)

ПОЛТАВА 2021 р.

ЗМІСТ

ВСТУП	3
1. ПОСТАНОВКА ЗАДАЧІ.....	5
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Структура дистанційного курсу	8
2.2. Використання тренажерів в дистанційному навчанні	11
3. ТЕОРЕТИЧНА ЧАСТИНА	14
3.1. Огляд матеріалу за темою роботи	14
3.2. Алгоритмізація задачі за темою роботи	22
3.3. Розробка блок-схеми, яка підлягає програмуванню	28
4. ПРАКТИЧНА ЧАСТИНА	31
4.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи.....	31
4.2. Опис процесу програмної реалізації	Ошибка! Закладка не определена.
4.3. Опис програми.....	Ошибка! Закладка не определена.
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	37
ДОДАТОК А. КОД ПРОГРАМИ	39

ВСТУП

Дистанційна освіта (ДО) на базі інформаційних технологій має інтернаціональний і глобальний характер. Студенти отримують можливість доступу до великого об'єму текстової, аудіо- та відеоінформації, проходження тестування та представлення виконаних робіт, спілкування між собою на значній відстані.

Дистанційна освіта — це міжнародний термін, який трактується як освіта на відстані та означає цілеспрямоване та методично організоване керівництво навчально-пізнавальною діяльністю осіб, які знаходяться на відстані від навчального закладу та не вступають у постійний контакт з викладацьким складом. на основі цього визначення ДО можна трактувати як самоосвіту, яка передбачає зворотній зв'язок із викладачем (навчальним закладом).

Важливою обставиною при впровадженні дистанційної форми освіти є визначення якості навчання. Під якістю освіти розуміється така організація взаємодії викладача і студентів, тобто навчального процесу, яка відповідає основним принципам концепції навчання, запитам сучасного суспільства та прогнозованих компетенції, якими має володіти випускник навчального закладу, щоб бути конкурентоспроможним фахівцем.

Метою роботи є розробка тренажеру з теми «Математичні основи теорії програмування» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є процес дистанційного навчання математичним дисциплінам.

Предметом розробки є програма тренажеру з теми «Математичні основи теорії програмування».

Перелік використаних методів — застосування математичних основ теорії програмування, мова програмування Java.

Тренажер готовий до використання в дистанційному курсі «Теорія програмування».

Робота складається з чотирьох розділів.

Обсяг пояснювальної записки: 40 стор., в т.ч. основна частина - 35 стор., джерела - 9 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Навчальний комп'ютерний інтерактивний тренажер – це програма, призначена для самостійного вивчення (або повторення) з одночасним контролем знань із певної теми. Всі інтерактивні завдання в такому тренажері повинні припускати наявність зворотного зв'язку, можливості корекції дій і можливості здійснювати практичні дії. До інтерактивних завдань у тренажері можна віднести послідовності питань з інтерактивними підказками і практикуми.

Завдяки доступності засобів створення тренажерів, великого вибору програмних комплексів, тренажери в сучасній освіті займають важливе місце при формуванні та закріпленні знань, умінь і навичок того, хто навчається й виконують роль педагогічного інструменту, що дозволяє підвищити якість освітнього процесу.

Упровадження навчальних комп'ютерних інтерактивних тренажерів сприяє оптимальному вирішенню завдань і усуненню низки недоліків традиційного способу навчання, а саме дозволяє:

- ініціювати чималий інтерес у здобувачів вищої освіти поряд із доступністю для них, тим самим підвищити активність і самостійність їх навчальної роботи;
- привернути увагу здобувачів вищої освіти, ураховуючи їх психологічні особливості, поліпшити сприйняття навчального матеріалу за рахунок його мультимедійності;
- забезпечити повний контроль засвоєння матеріалу кожним здобувачем вищої освіти;
- полегшити процес повторення і тренінгу при підготовці до іспитів та інших форм контролю знань;
- розвантажити викладачів від рутини контролю й консультування;

- використовувати позааудиторний час для вивчення конструкцій у вигляді домашніх завдань;
- поліпшити дистанційні форми навчальної роботи.

Ці питання в усій повноті можна вирішувати за допомогою інтерактивних навчальних комп'ютерних тренажерів, створених на комп'ютерах.

На рисунку 1.1 представлена принципова схема процесу навчання із застосуванням навчального комп'ютерного інтерактивного тренажеру.

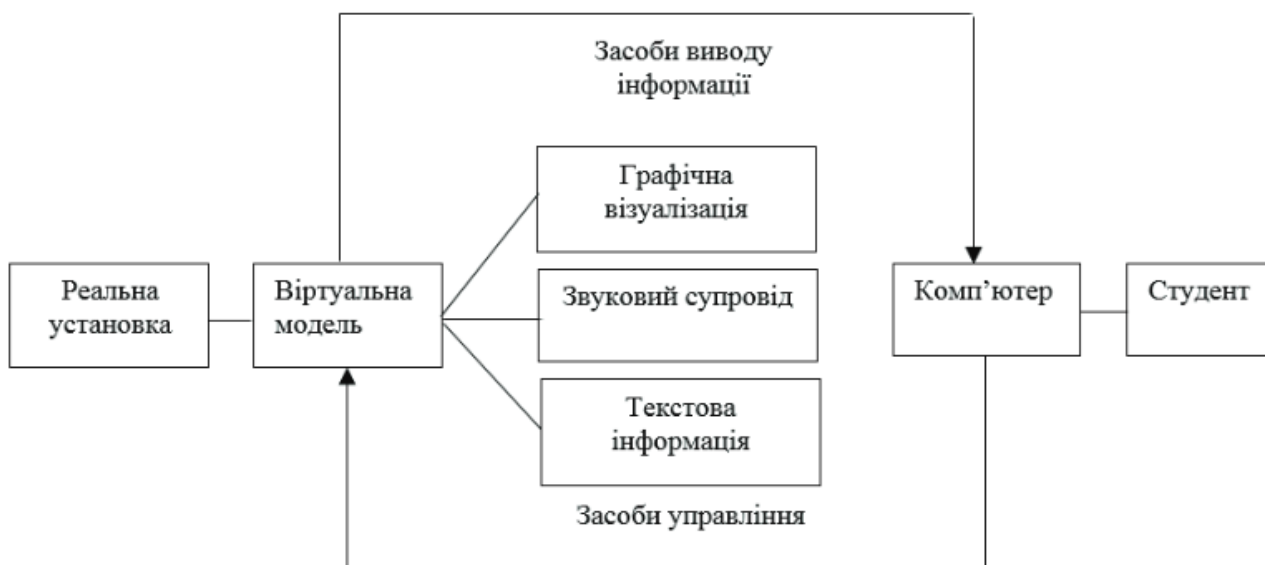


Рисунок 1.1 – Принцип дії тренажеру

Апаратні можливості тренажеру – це сучасний персональний комп'ютер, оснащений якісними пристроями введення/виведення інформації. Програмні засоби – це математично обґрунтована віртуальна модель, що включає в себе систему графічної візуалізації, звуковий супровід і текстову інформацію. Введення й виведення інформації здійснюється згідно з розробленим алгоритмом – програмного коду віртуальної моделі або за допомогою платформ дистанційного навчання.

Розглянемо основні етапи роботи тренажеру з теми «Математичні основи теорії програмування», який потрібно розробити для дистанційного курсу «Теорія програмування»:

- виведення основної інформації про тренажер;

- перехід до завдань, що будуть перевіряти знання з теми;
- виконання завдань та їх перевірка;
- виведення повідомлення про помилку у разі невірної відповіді;
- перехід до результатів проходження тренажеру;
- після проходження тренажеру надається можливість розпочати виконання завдань з самого початку.

Тренажер має мати не складний для користувача інтерфейс.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Структура дистанційного курсу

Дистанційне навчання – це запланована викладачем діяльність для засвоєння структурованої інформації.

Дистанційний навчальний процес – це відтворення традиційного навчального процесу засобами ІКТ (спілкування, співробітництво, співтворчість, самостійна робота тощо).

Ефективність будь-якого виду дистанційного навчання залежить від чотирьох складових:

- а) ефективної взаємодії викладача і учня, незважаючи на те, що вони фізично розділені відстанню;
- б) використання педагогічних технологій;
- в) ефективності розроблених методичних матеріалів і способів їх подання;
- г) ефективності зворотного зв'язку.

Метою навчання є та поведінка, ті знання, вміння та навички, які повинен продемонструвати студент, щоб його можна було назвати “компетентним”. Також це опис очікуваних результатів навчання, а не сам навчальний процес

При створенні дистанційних курсів необхідно враховувати наступні вимоги:

1. Мотивація.

Мотивація - необхідна складова навчання, яка повинна підтримуватися протягом всього процесу навчання. Велике значення має чітко визначена мета, яка ставиться перед студентом. Мотивація швидко знижується, якщо рівень поставлених завдань не відповідає рівню підготовки студента.

2. Постановка навчальної мети.

Студент з самого початку роботи за комп'ютером повинен знати, що від нього вимагається. Завдання навчання повинні бути чітко і ясно сформульовані в програмі.

3. Створення передумов до сприйняття навчального матеріалу.

Для створення передумов до сприйняття навчального матеріалу можуть бути корисні допоміжні матеріали (керівництва для студентів), що входять в комплект готового пакету або підготовлені самим викладачем. Можливе проведення попереднього тестування.

4. Подача навчального матеріалу.

Стратегія подачі матеріалу визначається залежно від розв'язуваних навчальних завдань. Важливою проблемою є оформлення кадрів, що подаються на екран монітора. Необхідно використовувати відомі принципи «сприйняття».

5. Зворотній зв'язок.

Цей критерій має ключове значення для учня, менше - в тестуючій програмі, більше - у тренажерній. Комп'ютер здатний забезпечувати зворотний зв'язок, причому допомога ця може бути індивідуальною.

6. Оцінка.

У ході роботи з комп'ютером студенти повинні знати, як вони справляються з навчальним матеріалом. Однак переважно не вказувати кількість неправильних відповідей до остаточного підбиття підсумків. Більшість студентів, як правило, стимулює невелика кількість завдань, велика кількість – стимулює менше. Найбільш важливим у дистанційному курсі є організація комунікацій "студент - викладач - студенти". Для цих цілей рекомендується організація роботи студентів у проектах або "навчання у співпраці", форуми.

На рисунку 2.1 зображено схематичний процес дистанційного навчання.



Рисунок 2.1 – Принцип дії тренажеру

Переваги дистанційної форми навчання:

- Свобода у часі. Не мають жорсткого графіка для виконання робіт.
- Свобода у просторі. Розташування не залежить від географічного розташування навчального закладу.

І найголовніше, студенти мають змогу працювати самостійно, індивідуально або у співробітництві, спілкуватись за будь-яких потреб і мати постійну підтримку і оцінку своєї діяльності з боку досвідченого викладача (тьютора).

2.2. Використання тренажерів в дистанційному навчанні

Інтерактивний інтерфейс, можливість зміни мови (українська-російська-англійська), отримання довідкової інформації, оперативного виправлення помилок, індивідуальний темп виконання завдання активізують пізнавальну діяльність студента. Створення тренажерів для задач, що мають чітку послідовність розв'язання, полягає у написанні сценарію (алгоритму), його програмної реалізації та апробації. Саме на етапі проектування закладаються основні функціональні можливості навчального електронного засобу: можливість спрощувати чи ускладнювати навчальну задачу, створення проблемної ситуації, призупинення процесу у будь який момент часу для аналізу рішення чи отримання допомоги викладача, неодноразове повторення для закріплення навичок тощо. Врахування інтервалів та кроку зміни параметрів забезпечують варіативність в кожному сеансі виконання тренажеру.

Такий автоматизований додаток продуктивно використовується студентом для напрацювання навичок вирішення типових математичних задач. Для засвоєння матеріалу навчальної дисципліни на діагностичному рівні передбачено використання напівавтоматичних тренажерів, де стандартні розрахунки на певних кроках зараховуються автоматично, а остаточні висновки перевіряються викладачем.

Одним із аспектів застосування тренажерів в навчальному процесі є врахування міжпредметних зв'язків, коли раніше вивчений матеріал може бути корисним для вирішення задач інших дисциплін. В опорній дисципліні тренажер є навчальним або контролюючим засобом, а в паралельній або перспективній виступає як обчислювальний засіб для розв'язання певних професійних завдань. На рис. 2.2 проілюстровано інтеграцію поглядів на функції тренажерів.

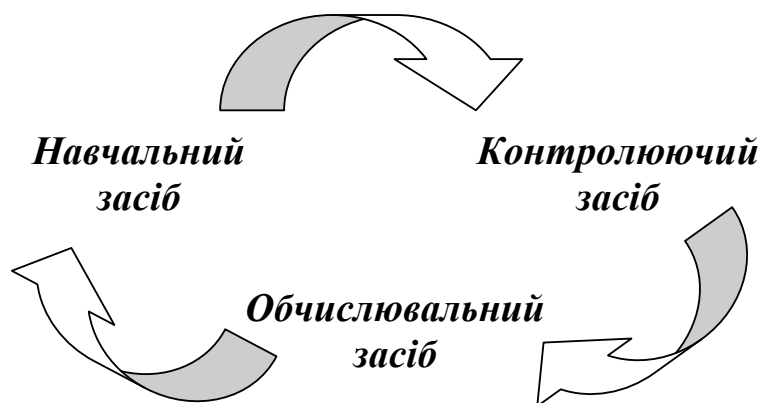


Рисунок 2.2 - Функції тренажерів

Врахування взаємозв'язків між дисциплінами дозволяє викладачам розробляти свої курси з опорою на раніше отримані знання студентів і планувати подачу матеріалу не ізольовано, а з урахуванням потреб подальших дисциплін. Формуючи сторінку з інформацією про дисципліну, її завдання, алгоритм вивчення автор-розробник вказує дисципліни, які для даної є базовими. В свою чергу, інші дистанційні курси посилатимуться на неї, як на базову. Таким чином формується реальна динамічна схема, що дозволяє наочно демонструвати міжпредметні зв'язки.

Продемонстрований підхід до врахування міжпредметних зв'язків в дистанційних курсах реалізує системність навчання і розуміння місця дисципліни у структурно-логічній схемі. Це дозволяє викладачам розробляти спільні комплексні проекти, вдосконалювати методичне забезпечення.

Практичні завдання є основним типом навчальних об'єктів, що застосовуються для організації практичної роботи та виконання індивідуальних завдань. Практичне завдання виконується студентом індивідуально та передбачає звіт про результати роботи, що надсилається студентом та перевіряється і оцінюється викладачем.

Метою практичних завдань є детальний розгляд студентами окремих теоретичних положень навчальної дисципліни та формування вмінь і навичок їх практичного застосування.

Проблемні або відкриті завдання в математичних дисциплінах найчастіше представлені постановкою розрахункової задачі з опорною схемою відповіді, методичними рекомендаціями до виконання і варіантами завдань для забезпечення індивідуального виконання студентом. В деяких випадках доцільна колективна форма роботи, що може бути реалізовано через чати, форуми або у вигляді мережових ігор.

При використанні стандартного або спеціалізованого програмного забезпечення (додатки MICROSOFT OFFICE, система MathCAD, пакет прикладних програм Matlab тощо) в практичній частині курсу наводяться приклади вирішення задач з використанням вказаної програми [2].

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Огляд матеріалу за темою роботи

Множина – сукупність об'єктів, які відрізняються за деякою ознакою.

Множини найчастіше записуються у вигляді:

а) $A = \{a, b, c\}$ – перерахування конкретних елементів;

б) $B = \{x \mid x > 0\}$ – опис характеристичної властивості елементів множини.

Порожню множину будемо позначати через \emptyset або $\{\}$, універсальну множину – через U . Універсальна множина містить всі елементи деякої предметної області. Наприклад, у теорії чисел це множина усіх цілих або натуральних чисел, у геометрії – множина усіх точок n -вимірного геометричного простору.

Відношення між множинами і елементами множин будемо записувати у вигляді:

- $a \in A$ – a належить A ;
 - $B \subseteq C$ – B є підмножиною C , B включається (включене) у C ;
 - $B \subset C$ – B є власною підмножиною C , B строго включається в C .
- \emptyset і C вважаються невластивими підмножинами множини C .

Множину всіх підмножин множини A називають булеаном множини A і позначають $P(A)$.

Операції над множинами.

Об'єднання: $A \cup B = \{x \mid x \in A \text{ або } x \in B\}.$

Перетинання: $A \cap B = \{x \mid x \in A \text{ і } x \in B\}$

Доповнення: $\overline{A} = \{x \mid x \in U \text{ і } x \notin A\}$

Різниця: $A \setminus B = \{x \mid x \in A \text{ і } x \notin B\}.$

Симетрична різниця: $A \oplus B = \{x \mid x \in (A \setminus B) \text{ або } x \in (B \setminus A)\}.$

Декартовим добутком $A \times B$ множин A і B називається множина усіх пар вигляду (a_i, b_j) , де $a_i \in A, b_j \in B$

Декартів добуток $A_1 \times A_2 \times \dots \times A_n$ множин A_1, A_2, \dots, A_n є множина усіх n -к вигляду $(a_{1i}, a_{2j}, \dots, a_{nk})$, де $a_{1i} \in A_1, \dots, a_{nk} \in A_n$.

Якщо $A_1 = A_2 = \dots = A_n = A$, декартів добуток називають **ступенем** A^n . При цьому $A^0 = \emptyset$, $A^1 = A$.

Приклади.

$$1 \quad A = \{a, b\}; B = \{c, d\}; A \times B = \{(a, c), (b, c), (a, d), (b, d)\}$$

2 A – відрізок на осі x , B – відрізок на осі y .

$A \times B$ – прямокутник на площині xy .

3 X – множина координат x , Y – множина координат y , s – множина довжин, a – множина кутів.

$X \times Y \times s \times a$ – множина усіх векторів на площині.

Елементи множин A і B знаходяться у **взаємо-однозначній відповідності**, якщо кожному $a \in A$ за деяким законом зіставлений один і той самий елемент $b \in B$, причому кожен $b \in B$ виявляється зіставленим одному і тільки одному $a \in A$.

Множини A і B **рівнопотужні** (або еквівалентні), якщо можна установити взаємооднозначну відповідність їхніх елементів.

Приклади: множина $N = \{1, 2, 3, \dots\}$ натуральних чисел еквівалентна множині $N^{(2)} = \{1, 4, 9, \dots\}$ квадратів натуральних чисел, множина цілих чисел еквівалентна множині парних чисел.

Таким чином, нескінченна множина може бути еквівалентною своїй власній підмножині.

Множини, еквівалентні множині натуральних чисел N , називаються **зліченими**.

Об'єднання скінченної або зліченної сукупності злічених множин також є зліченною множиною.

Приклад. Множина усіх слів скінченної довжини в алфавіті a_1, a_2, \dots, a_m – зліченна.

Множина усіх дійсних чисел інтервалу $(0,1)$ незліченна.

Множина, еквівалентна множині всіх дійсних чисел інтервалу $(0,1)$, називається множиною потужності **континуума**.

Множина усіх підмножин деякої зліченної множини є множиною потужності континуума.

Нехай $F(A)$ – множина усіх слів у скінченному алфавіті A . Будь-яка підмножина $L \subseteq F(A)$ називається мовою над алфавітом A .

Множина усіх мов над скінченним алфавітом A є множиною потужності континуума.

Потужність множини задається кардинальним числом. Кардинальне число зліченної множини позначають через \aleph_0 .

Множини найбільшої потужності не існує.

Візьмемо довільні множини A_1, A_2, \dots, A_n не обов'язково різні.

Під **n -арним**, або **n -відношенням**, на множинах A_1, A_2, \dots, A_n розуміють закон, що виділяє у декартовому добутку деяку підмножину

$$\rho^n \subseteq A_1 \times A_2 \times \dots \times A_n.$$

Часто під відношенням розуміють саму підмножину ρ^n .

Відношення будемо позначати грецькими буквами $\rho, \alpha, \beta, \gamma, \dots$ і спеціальними символами $<, =, >, \leq, \geq$.

Відношення ρ^n можна представити у матричній формі:

$$\rho^n = \left\| \begin{array}{ccc} a_1^{(1)} & . & a_1^{(n)} \\ . & . & . \\ a_r^{(1)} & . & a_r^{(n)} \end{array} \right\|.$$

Рядки матриці при цьому називають векторами відношення.

ρ^1 на A – унарне відношення, ρ^2 на A, B – бінарне, ρ^3 на A, B, C – тернарне.

Унарні відношення

Унарне відношення ρ^1 на множині A є характеристичною властивістю деякої підмножини.

Приклади відношень на N : $\alpha^1 = \{3, 7\}$; $\beta^1 = \{n \mid n > 5\}$;
 $\gamma^1 = \{n \mid n - \text{парне}\}$.

Множина всіх унарних відношень на A збігається з множиною всіх підмножин A .

Бінарні відношення

Якщо $a \subseteq A$ і $b \subseteq B$ знаходяться у відношенні ρ , це іноді записують так: $a\rho b$.

Бінарні відношення часто представляють за допомогою таблиць.

Нехай є відношення ρ на $A = \{a_1, \dots, a_r\}$ і $B = \{b_1, \dots, b_s\}$. Рядки таблиці відповідають елементам a_i , стовпці – елементам b_j , наявність відношення $a_i \rho b_j$ позначається як “*”.

ρ	2	3	5
2	*		*
3		*	

Тернарне відношення

Наприклад, з кожною бінарною операцією $F(x, y)$, зокрема “+”, “–”, можна зв'язати тернарне відношення $\varphi^3(x, y, z)$, для якого $z = F(x, y)$.

Приклади n-арних відношень

1. $A = (2, 3)$, $B = \{2, 3, 4, 6\}$, a і b знаходяться у відношенні ρ , якщо a є дільник b .

$$\rho = \begin{vmatrix} 2 & 2 \\ 2 & 4 \\ 2 & 6 \\ 3 & 3 \\ 3 & 6 \end{vmatrix}$$

2. Бінарне відношення $<$ на N :

$$<_N = \begin{vmatrix} 1 & 2 \\ 1 & 3 \\ \cdot & \cdot \\ 2 & 3 \\ 2 & 4 \\ \cdot & \cdot \end{vmatrix}$$

3. Відношення $<, =, >$ на множині слів.

4. Відношення ψ , що зіставляє кожному слову з $F(A)$ його код $q \in M$. Тут A – алфавіт; M – деяка множина.

Відношення широко використовуються при описі синтаксису мов і при перекладі з однієї мови на іншу.

$$\text{Нехай знову } \rho^n = \begin{vmatrix} a_1^{(1)} & \cdot & a_1^{(n)} \\ \cdot & \cdot & \cdot \\ a_r^{(1)} & \cdot & a_r^{(n)} \end{vmatrix}.$$

Проекцією $Pr^{(j)}(\rho^n)$ відношення ρ^n на множину A_j називається множина всіх елементів j -го стовпця матриці.

Перетином $S^{(j)}(\rho^n)$ відношення ρ^n за елементами $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ називається множина всіх елементів $a_{i_q}^{(j)}$, для яких n -ка $a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ належить даному відношенню.

Фактор-множиною A_j / ρ^n множини A_j стосовно ρ^n називається множина перетинів цього відношення за різними сукупностями $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$.

Приклад. Нехай відношення α^3 на множинах $A = \{a_1, a_2\}$, $B = \{b_1, b_2, b_3\}$, $C = \{0, 1\}$ визначається рівністю

$$\alpha^3 = \left\| \begin{array}{ccc} a_1 & b_1 & 0 \\ a_1 & b_2 & 0 \\ a_2 & b_1 & 1 \\ a_2 & b_2 & 1 \\ a_2 & b_3 & 0 \end{array} \right\|.$$

Тоді

$$\begin{aligned} \text{Пр}^1(\alpha^3) &= A, \text{Пр}^2(\alpha^3) = \{b_1, b_2\}, \text{Пр}^3(\alpha^3) = C; S_{b_2 0}^1(a^3) \\ &= \{a_1, a_2\}, S_{a_1 0}^2(a^3) = \{b_1, b_2\}. \end{aligned}$$

Фактор-множина C / α^3 подається у вигляді

$$\{S_{a_1 b_1}^3(a^3) = \{0\}, S_{a_1 b_2}^3 = \{1\}, S_{a_2 b_2}^3 = C, S_{a_1 b_3}^3 = \emptyset, S_{a_2 b_3}^3 = \emptyset\}.$$

Відношення φ^n на множинах A_1, A_2, \dots, A_n називається функціональним, якщо для кожної послідовності $a_i^{(1)}, \dots, a_i^{(n-1)} \in A_1 \times \dots \times A_{n-1}$ перетин S^n за цими елементами містить не більше ніж один елемент $a_i^{(n)} \in A_n$.

Якщо перетин містить точно один елемент $a_i^{(n)}$, то φ^n називають всюди визначеним функціональним відношенням.

Приклади. Нехай є множини $A = \{a_1, a_2\}$, $B = \{b_1, b_2\}$, $C = \{0, 1\}$

Відношення

$$\alpha^3 = \begin{pmatrix} a_1 & b_1 & 0 \\ a_1 & b_2 & 0 \\ a_2 & b_1 & 1 \end{pmatrix}, \quad \beta^3 = \begin{pmatrix} a_1 & b_1 & 1 \\ a_1 & b_2 & 0 \\ a_2 & b_1 & 0 \\ a_2 & b_2 & 1 \end{pmatrix}$$

– функціональні, причому β^3 – усюди визначене.

З функціональним відношенням φ^n зв'язують функцію $F(x_1, \dots, x_{n-1})$. У вищенаведеному прикладі з α^3 зв'язана часткова функція $F(x, y)$:

$F(a_1, b_1) = F(a_1, b_2) = 0, F(a_2, b_1) = 1$, а з β^3 зв'язана функція $F(x, y)$:

$$F(a_1, b_1) = F(a_2, b_2) = 1, F(a_1, b_2) = F(a_2, b_1) = 0.$$

Нехай φ – бінарне відношення на множинах A, B .

Відношення φ називається відображенням множини A в B , якщо φ функціональне й усюди визначене.

Відображення φ множини A у B називається відображенням A на B , якщо кожний елемент $b \in B$ має в A хоча б один прообраз.

Бінарне відношення ρ на якійсь множині A , яке задовольняє властивості рефлексивності, транзитивності й антисиметричності, називається відношенням **часткового порядку**.

Відношення порядку символічно позначається через \leq .

Множина A з визначеним на ньому відношенням часткового порядку називається **частково упорядкованою**.

Елементи $a, b \in A$ такі, що $a\rho b$ або $b\rho a$ називаються порівнянними елементами частково упорядкованої множини A .

Частково упорядкована множина, у якій кожна пара елементів порівнянна, називається **лінійно упорядкованою множиною**, або **ланцюгом**.

Нехай A – частково упорядкована множина; B – його підмножина. **Верхньою гранню** B у множині A називається елемент $a \in A$, такий, що $b \leq a$ для будь-якого $b \in B$. Елемент a – **найбільший** у множині A , якщо a – верхня грань самого A . Елемент $m \in A$ називається **максимальним** у множині A , якщо не існує $b \in A$, такого, що $m < b$.

Найбільший елемент множини A є єдиним максимальним елементом даної множини. Множина A може не мати найбільшого елемента і в той же час мати один або кілька максимальних.

Аналогічні міркування справедливі для нижньої грані.

Лінійно впорядкована множина A називається **повністю впорядкованою**, якщо кожна непорожня підмножина $B \subseteq A$ має найменший елемент.

Множина N є повністю впорядкованою, множина Z – ні.

Нехай B – підмножина частково впорядкованої множини A .

Верхнім конусом B^Δ множини B називається сукупність усіх верхніх граней даної множини, **точною верхньою гранню** B у множині A – елемент $a \in B^\Delta$, такий, що $a \leq b$ для будь-якого $b \in B^\Delta$.

Частково упорядкована множина S називається **структурою**, або решіткою, якщо для будь-якої пари елементів $a, b \in S$ існує точна нижня грань і точна верхня грань.

Дистрибутивна структура S з елементами 0 і 1 називається **булевою структурою**, або булевою алгеброю, якщо для будь-якого елемента $a \in S$ існує елемент $\bar{a} \in S$ такий, що $a \cap \bar{a} = 0$ і $a \cup \bar{a} = 1$. Елемент \bar{a} називається доповненням елемента a у структурі S .

Булевою структурою є множина усіх мов над алфавітом A [3-5].

3.2. Алгоритмізація задачі за темою роботи

Крок 1. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Множина – це:»

Варіанти відповіді:

- об'єкти, які відрізняються за деякою ознакою;
- сукупність об'єктів, які відрізняються за деякою ознакою;
- сукупність об'єктів, які схожі за деякою ознакою.

Якщо вибрано другий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 2. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Множини найчастіше записуються у вигляді перерахування конкретних елементів або опису характеристичної властивості елементів множини. Вкажіть перерахування конкретних елементів:»

Варіанти відповіді:

- $A = \{a, b, c\}$;
- $B = \{x \mid x > 0\}$.

Якщо вибрано перший варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 3. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Множини найчастіше записуються у вигляді перерахування конкретних елементів або опису характеристичної властивості елементів множини. Вкажіть опис характеристичної властивості елементів множини:»

Варіанти відповіді:

- $A = \{a, b, c\}$;
- $B = \{x \mid x > 0\}$.

Якщо вибрано другий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 4. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Відношення між множинами і елементами множин будемо записувати у вигляді $a \in A$:»

Варіанти відповіді:

- a належить A ;
- A належить a .

Якщо вибрано перший варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 5. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Відношення між множинами і елементами множин будемо записувати у вигляді $B \subseteq C$:»

Варіанти відповіді:

- B є власною підмножиною C , B строго включається в C ;
- B є підмножиною C , B включається (включене) у C .

Якщо вибрано другий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 6. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Відношення між множинами і елементами множин будемо записувати у вигляді $B \subset C$:»

Варіанти відповіді:

- B є власною підмножиною C , B строго включається в C ;
- B є підмножиною C , B включається (включене) у C .

Якщо вибрано перший варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 7. Користувачу виводиться завдання: «Вкажіть правильну відповідь в комірці.

Операції над множинами: Об'єднання ____ = $\{x \mid x \in A \text{ або } x \in B\}$ »

Варіанти вибору:

- $A \cup B$;
- $A \cap B$;
- $A \oplus B$;
- $A \setminus B$;
- \bar{A} .

Якщо вибрано перший варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 8. Користувачу виводиться завдання: «Вкажіть правильну відповідь в комірці.

Операції над множинами: Перетинання ____ = $\{x \mid x \in A \text{ і } x \in B\}$ »

Варіанти вибору:

- $A \cup B$;
- $A \cap B$;
- $A \oplus B$;
- $A \setminus B$;
- \bar{A} .

Якщо вибрано другий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 9. Користувачу виводиться завдання: «Вкажіть правильну відповідь в комірці.

Операції над множинами: Доповнення ____ = $\{x \mid x \in U \text{ і } x \notin A\}$ »

Варіанти вибору:

- $A \cup B$;
- $A \cap B$;
- $A \oplus B$;

- $A \setminus B$;
- \overline{A} .

Якщо вибрано п'ятий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 10. Користувачу виводиться завдання: «Вкажіть правильну відповідь в комірці.

Операції над множинами: Різниця _____ = $\{x \mid x \in A \text{ і } x \notin B\}$ »

Варіанти вибору:

- $A \cup B$;
- $A \cap B$;
- $A \oplus B$;
- $A \setminus B$;
- \overline{A} .

Якщо вибрано четвертий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 11. Користувачу виводиться завдання: «Вкажіть правильну відповідь в комірці.

Операції над множинами: Симетрична різниця

_____ = $\{x \mid x \in (A \setminus B) \text{ або } x \in (B \setminus A)\}$ »

Варіанти вибору:

- $A \cup B$;
- $A \cap B$;
- $A \oplus B$;
- $A \setminus B$;
- \overline{A} .

Якщо вибрано третій варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 12. Користувачу виводиться завдання: «Вкажіть розв'язок прикладу.

Декартовим добутком $A \times B$ множин A і B називається множина усіх пар вигляду (a_i, b_j) , де $a_i \in A, b_j \in B$. **Декартів добуток** $A_1 \times A_2 \times \dots \times A_n$ множин A_1, A_2, \dots, A_n є множина усіх n -к вигляду $(a_{1i}, a_{2j}, \dots, a_{nk})$, де $a_{1i} \in A_1, \dots, a_{nk} \in A_n$. Якщо $A_1 = A_2 = \dots = A_n = A$, декартів добуток називають **ступенем** A^n . При цьому $A^0 = \emptyset$, $A^1 = A$.

Приклад 1. $A = \{a, b\}; B = \{c, d\}$ »

$$A \times B = \{ \rule{1.5cm}{0.4pt} \}$$

Якщо вказано $(a, c), (b, c), (a, d), (b, d)$, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 13. Користувачу виводиться завдання: «Вкажіть розв’язок прикладу.

Приклад 2. A – відрізок на осі x , B – відрізок на осі y .

$A \times B$ – прямокутник на площині _____»

Якщо вказано xu , то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 14. Користувачу виводиться завдання: «Вкажіть правильну відповідь в комірці.

Приклад 3. X – множина координат x , Y – множина координат y , s – множина довжин, a – множина кутів.

$X \times Y \times s \times a$ – множина усіх _____ на площині»

Варіанти вибору:

- координат;
- довжин;
- векторів.

Якщо вибрано третій варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 15. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Під n -арним, або n -відношенням, на множинах A_1, A_2, \dots, A_n розуміють закон, що виділяє у декартовому добутку деяку підмножину:»

Варіанти відповіді:

- $\rho^n \subseteq A_1 \times A_2 \times \dots \times A_n$;
- $\rho^n \subset A_1 \times A_2 \times \dots \times A_n$.

Якщо вибрано перший варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 16. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Проекцією $Pr^{(j)}(\rho^n)$ відношення ρ^n на множину A_j :»

Варіанти відповіді:

- множина всіх елементів n -го стовпця матриці;
- множина всіх елементів j -го стовпця матриці.

Якщо вибрано другий варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 17. Користувачу виводиться завдання: «Виберіть правильну відповідь.

Перетином $S^{(j)}(\rho^n)$ відношення ρ^n за елементами $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ називається:»

Варіанти відповіді:

- множина всіх елементів $a_{i_q}^{(j)}$, для яких n -ка $a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ належить даному відношенню;

- множина всіх елементів $a_q^{(i)}$, для яких n -ка $a_i^{(1)}, \dots, a_i^{(j-1)}, a_{i_q}^{(j)}, a_i^{(j+1)}, \dots, a_i^{(n)}$ належить даному відношенню.

Якщо вибрано перший варіант, то відбувається перехід на наступний крок. Інакше – відображається повідомлення про помилку.

Крок 18. Користувачу виводиться завдання: «Виберіть правильну відповідь».

Фактор-множиною A_j / ρ^n множини A_j стосовно ρ^n називається:»

Варіанти відповіді:

- множина перетинів цього відношення за різними сукупностями $a_i^{(1)}, \dots, a_i^{(j-1)}, a_i^{(j+1)}, \dots, a_i^{(n)}$;
- множина перетинів цього відношення за різними сукупностями $a_i^{(1)}, \dots, a_i^{(j)}, a_i^{(j)}, \dots, a_i^{(n)}$.

Якщо вибрано перший варіант, то відбувається перехід до результатів проходження тренажеру. Інакше – відображається повідомлення про помилку.

3.3. Розробка блок-схеми, яка підлягає програмуванню

На рисунку 3.1 зображена блок-схема роботи програми, на рисунку 3.2 – блок-схема алгоритму тренажеру.

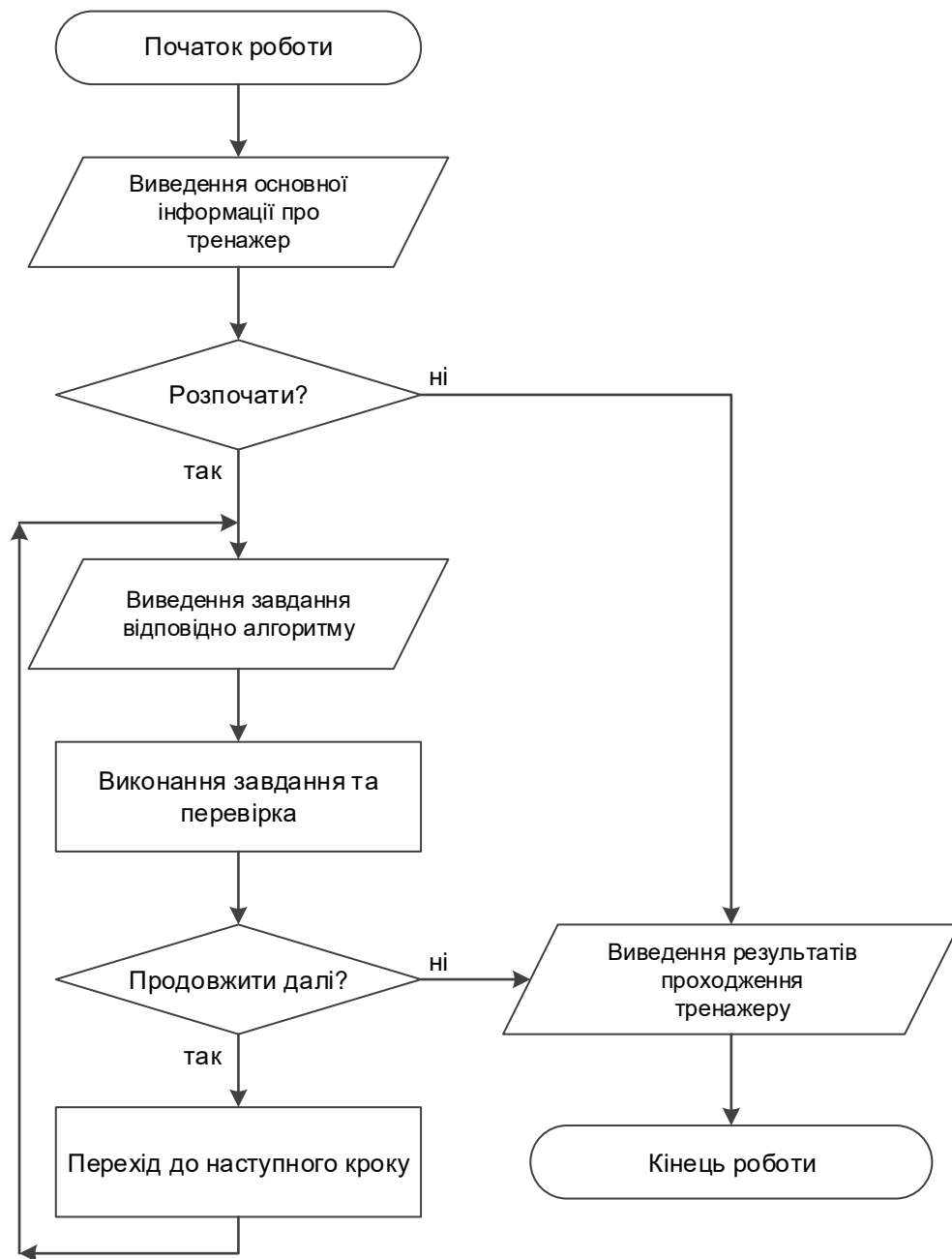


Рисунок 3.1 – Блок-схема роботи програми

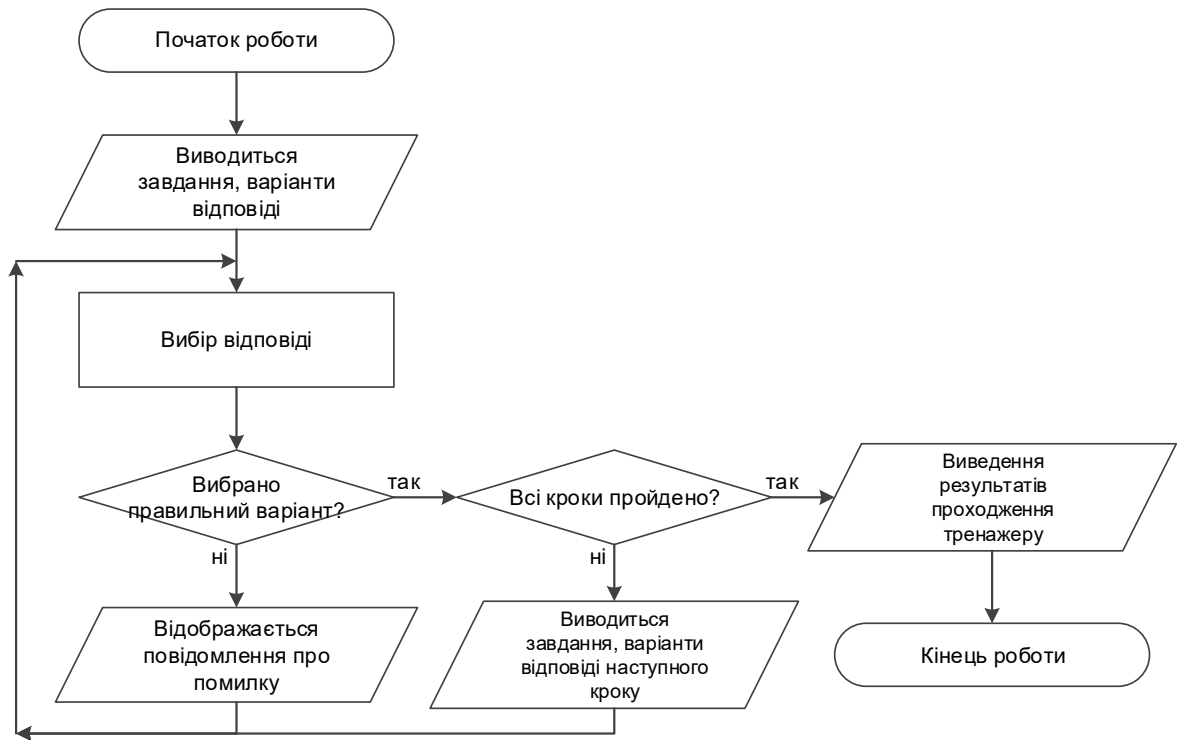


Рисунок 3.2 – Блок-схема алгоритму тренажеру

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи

Java — об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems. Програми Java зазвичай компілюються в спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині (JVM) незалежно від комп'ютерної архітектури. Дата офіційного випуску - 23 травня 1995 [6].

Перевага 1 - Java легко вивчається.

Синтаксис цієї мови програмування схожий на звичайну англійську мову. В ній є мінімальна кількість складних для запам'ятовування символів. По суті, після встановлення JDK, настройки PATH і вивчення особливостей Classpath спеціаліст вже може створювати елементарні програми на Java.

Перевага 2 - У Java прийняті концепції хорошого програмування.

Java — об'єктно-орієнтована мова, причому це саме «об'єктне» в Java реалізоване просто-напросто на відмінно! Разом з ООП Ви вивчите концепції наслідування, абстракції, поліморфізму і так далі. Java навчить концепціям, які можна застосовувати в більшості інших мов, наприклад, в Python

Перевага 3 - Java така мова програмування рідко змінюється.

Для новачків це ВЕЛИКИЙ плюс, що не потрібно відволікатися і постійно слідкувати за новими тенденціями ІТ. Новий функціонал Java інколи навіть занадто довго вводиться... наприклад, подібне відбувалося із замиканнями.

Перевага 4 - Користувачі Java мають доступ до великої колекції бібліотек з відкритим кодом.

Найкращі програмісти світу розробили шаблони, які роблять процедуру розробки більш простою. Підтримка з боку таких гігантів, як

Google і Apache, дає зрозуміти, що ця мова програмування ще довго буде використовуватися в проектах. Java має гарно пророблений API, великий вибір інструментарію, велику кількість фреймворків.

Перевага 5 - Java має гігантське співтовариство по всьому світу.

Існує маса форумів, тематичних порталів та інших ресурсів, де обмінюються знаннями користувачі Java. Там спілкуються досвідчені програмісти, переймають досвід новачки, обговорюють новини, пов'язані з мовою програмування. Ви можете бути впевнені в тому, що ніяких проблем з пошуком інформації точно не буде.

Велика частина сучасних стартапів не використовує Java, так як на даний момент існують більш швидкі шляхи виконання того ж обсягу роботи. Проте ми навчаємо вивчати те, що приносить найбільше задоволення, так Ви станете справжнім експертом в обраній справі, зокрема кодуванні на улюбленій мові програмування [7].

Створення графічного інтерфейсу на Java: саме вікно з меню і кнопками, а також його поведінку будуть створюватися за допомогою бібліотеки Swing. Промальовування - за допомогою бібліотеки AWT (Точніше, Java 2D). Це два основних пакети для створення графічних інтерфейсів на Java.

Робота AWT спочатку заснована на так званих peer-інтерфейсів. Суть полягає в тому, що при необхідності виведення на екран об'єкта Java, операційною системою створюється парний йому графічний об'єкт, який, власне, і відображається. Ці два об'єкти взаємодіють між собою під час роботи програми. Така реалізація призводить до того, що для кожної платформи доводиться випускати власний JDK.

Пізніше в AWT були створені компоненти, які не використовують peer-інтерфейси - «легкі» (lightweight) компоненти. Бібліотека цих компонентів була названа Swing. Тобто Swing, по суті, є розширенням AWT.

Сама ж AWT була доповнена новими засобами малювання і виведення зображень, які отримали назву Java 2D.

Базовим будівельним блоком всієї бібліотеки візуальних компонентів Swing є JComponent. Це суперклас кожного компонента. Він є абстрактним класом, тому в дійсності ви не можете створити JComponent, але він містить буквально сотні функцій, які кожен компонент Swing може використовувати як результат ієрархії класів.

JFrame (Вікно додатка) - основний контейнер, що дозволяє додавати до себе інші компоненти для їх організації та надання користувачеві. JFrame виступає в якості моста між незалежними від ОС Swing-частинами і реальної ОС, на якій вони працюють. JFrame реєструється як вікно в ОС і таким чином отримує багато із знайомих властивостей вікна операційної системи.

JMenu / JMenuItem / JMenuBar - призначені для розробки системи меню в JFrame. Основою будь-якої системи меню є JMenuBar, кожен JMenu і JMenuItem створюється з ним. JMenu є підкласом JMenuItem. Однак за зовнішнім виглядом вони мають відмінність: JMenu використовується для утримання інших JMenuItem і JMenu; JMenuItem при виборі активізує дію.

JLabel (Мітка) - призначений для опису (текстового або графічного) інших елементів.

JButton (Кнопка) - основний активний компонент, що дозволяє виконати будь-які дії при її натисканні. Крім стандартних методів, які керують відображенням компонента, містить групу методів для управління своїм станом (активна / неактивна, обрана / не вибрав, мишка зверху / мишки немає, натиснута / віджата).

JTextField (Текстове поле) - дозволяє користувачеві вводити текстові дані, які можуть бути оброблені в програмі.

JTextArea розвиває JTextField, дозволяючи вводити кілька рядків.

JPasswordField (Поле для введення пароля) - різновид JTextField, що дозволяє приховувати. Зауважте, що ви.

JComboBox (Комбінований список) - дозволяє користувачеві вибрати елемент з існуючого списку (або додати до списку новий елемент).

JCheckBox (Прапорець) і JRadioButton (Перемикач) - надають користувачеві варіанти для вибору. JRadioButton зазвичай групуються разом для надання користувачеві питання з примусовим відповіддю (відповіді взаємовиключні - може бути тільки одна відповідь на питання). Як тільки ви вибрали JRadioButton, ви не можете зняти його позначку до тих пір, поки не виберете інший варіант з групи. JCheckBox працює інакше. Він дозволяє відзначати / знімати позначку з варіанту в будь-який час і вибирати кілька відповідей на питання. Класом, який дозволяє групувати разом компоненти JCheckBox або JRadioButton, є клас ButtonGroup.

JSlider - елемент для вибору числового значення з графічно представленого діапазону.

JSpinner - призначений для вибору з групи значень. У цьому він аналогічний JComboBox, хоча їх використання не повинно бути взаємозамінні. Ви повинні використовувати JSpinner тільки для логічно послідовних варіантів - числа і дати ідеально підходять для цього. JComboBox, з іншого боку, більш хороший вибір для подання здаються випадковими варіантів, які не мають взаємозв'язку між собою.

JToolBar виступає як контейнер для інших компонентів (JButtons, JComboBoxes і т.д.), які спільно утворюють панелі інструментів, що зустрічаються в більшості додатків. Панелі інструментів дозволяють програмі розміщувати часто використовувані команди в зручному місці і групувати їх. Зазвичай кнопки панелі інструментів відповідають командам в меню.

JToolTip - це невеликі «бульбашки», які спливають, коли ви наводите і тримаєте курсор мишки над чим-небудь. Вони можуть бути дуже корисні в додатках, надаючи підказки для елементів, деталізуючи інформацію або навіть показуючи повний текст елемента в стислих UI. Вони активізуються в Swing, якщо залишити курсор мишки над компонентом на певну кількість часу; зазвичай вони з'являються приблизно через секунду після зупинки

мишки і залишаються видимими до тих пір, поки курсор залишається над компонентом.

JOptionPane - клас для надання UI-розробникам способу видачі простих повідомлень (про помилку або іншої інформації) і швидкого отримання даних (наприклад, імені або номера).

JScrollPane - Swing-компонент для обробки всіх дій по прокручуванні.

JList є корисним компонентом для надання користувачеві багатьох варіантів для вибору. Ви можете представляти його як розширення JComboBox. JList надає більше варіантів і додає можливість вибору декількох варіантів. Вибір між JList і JComboBox часто полягає в наступному: якщо вам потрібна можливість множинного вибору або є більше 15 варіантів (хоча це число не є загальним правилом), ви повинні завжди вибирати JList. Ви повинні використовувати JList спільно з JScrollPane, оскільки він може надавати більше варіантів, ніж поміщається в видимій області. JList має також модель вибору, яку ви можете встановити в різні типи вибору варіантів. Такими типами є: одиночний вибір (ви можете вибрати тільки один варіант), одиночний інтервал (ви можете вибрати суміжні варіанти, але в будь-якому їх кількості) і множинний інтервал (ви можете вибрати будь-яке число варіантів в будь-яких комбінаціях) [6, 8].

ВИСНОВКИ

Перед розробкою алгоритму було розглянуто структуру дистанційного курсу, використання тренажерів в дистанційному навчанні.

При розробці алгоритму було розглянуто основні означення з таких тем:

- Множини;
- Відношення;
- Порядок, структури.

Розглянемо основні етапи роботи тренажеру з теми «Математичні основи теорії програмування», який потрібно розробити для дистанційного курсу «Теорія програмування»:

- виведення основної інформації про тренажер;
- перехід до завдань, що будуть перевіряти знання з теми;
- виконання завдань та їх перевірка;
- виведення повідомлення про помилку у разі невірної відповіді;
- перехід до результатів проходження тренажеру;
- після проходження тренажеру надається можливість розпочати виконання завдань з самого початку.

Віртуальні тренажери надають неоціненну допомогу як студенту (вони дозволяють наочно подати матеріал, повторювати певні дії до повного розуміння і закріплення вивченого матеріалу), так і викладачеві (не вимагають постійної присутності, неупереджено оцінюють отримане рішення, методичні рекомендації видаються автоматично, програма сама вказує на допущені помилки).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації щодо оформлення пояснювальних записок до курсових проектів (робіт) для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» галузь знань - 12 «Інформаційні технології» / О. О. Ємець – Полтава : РВВ ПУЕТ, 2017. – 69с.
2. Шовкопляс О.А. Звіт про науково-дослідну роботу «Моделювання навчального процесу вивчення економіко-математичних дисциплін з використанням комп'ютерних технологій» / О. А. Шовкопляс, О. О. Базиль, О. А. Літвіненко, О. С. Мазманішвілі – Суми: Вид-во СумДУ, 2015. – 53 с.
3. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напряму 6.040302 «Інформатика» – Режим доступу: <http://tprogr.ho.ua>.
4. Бабій М.С. Теорія програмування: Навчальний посібник [Електронний ресурс] / М.С. Бабій, О.П. Чекалов.– Суми: Вид-во СумДУ, 2009. – 181 с.
5. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://tp.unicyb.kiev.ua/doc/TOP.pdf>.
6. Шилдт Герберт Java 8. Полное руководство; 9-е изд.: Пер. с англ. / Герберт Шилдт - М. : ООО "И.Д. Вильямс", 2015. - 1376 с.
7. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Кей Хорстманн - М. : ООО "И.Д. Вильямс", 2018. - 864 с.
8. Машнин Тимур Графические интерфейсы пользователя Java / Тимур Машнин - Litres, 2021. - 956 с.

9. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.

ДОДАТОК А. КОД ПРОГРАМИ**ДОДАТОК А**